CMS Event Display Program

- GEANT3 Physics
- PHIGS Graphics
- MOTIF GUI
1. Full 3D control of detector orientation
2. Preset views for often used magnifications and orientations
3. Z-Slice
4. Independent or ganged x and y magnification
5. Text annotation widget
6. Vector Postscript output (either direct or by CGM metafile to PS utility)
7. Visibility control of sub-detector and sub-event elements
8. Simultaneous multiple views
9. Picking
Design Decisions

• Beg, Borrow, or Steal (but only good stuff)
  – Use existing HEP Physics libraries (implies FORTRAN code will be present)
  – Use industry standard graphics package (PHIGS)
  – Use a standard unix GUI (Motif)

• Create Generic Structure Definition Format
  – Describe possible scene as directed acyclic graph: the frame
  – Create for nodes when first requested
  – Advantages:
    * Follows changes in detector organization w/o re-coding
    * Creates selective visibility widget
    * Doesn’t build what isn’t requested

Modularity of Design Reflected in Modular Controls
Integrating & Customizing

• Physics: implement detector in GEANT
• Graphics...
  – Remove calls to HIGZ at object (3D) level
  – Replace with PHIGS calls

  ![Diagram showing the integration process between GEANT and PHIGS]

• Interface: Custom Design
  – major branch of display tree
    ⇒ independent widget
  – Pre-set views:
    – Logo
• Tuning
Details of Tuning GEANT3

- Streamlined GEANT3 tree-following routine: too slow
- Division of information among GDRAWS, shape drawing routines, line primitives was fortuitous
- GMT
- Re-implement only essential shapes

Lessons to take onward...

- Acceptable performance requires entire program be responsive, not just graphics section
- Lucky happenstance (foresight from designers?) that extension from 2D to 3D graphics was relatively simple - required modularity and open access to internals
- HIGZ was meant to be API for graphics but entire API was pulled
- Full complexity of shapes as implemented by physicists was impossible to use. Required close co-ordination plus hand-tuning to extract graphical content.
Interface Details

Tried to be generic but human nature requires acknowledgment

- Label magnifications with tags specific to the detector
- Set dimensions on sliders to values appropriate for detector
- Specific help instructions
- No cryptic icons
- Breakout of Frame to first branching
- Macros for editing Postscript output (try that on a bitmap...)
Performance & Maintenance Issues

• Desired characteristics:
  – Fast: should respond to the user w/o hesitation
  – Uniform: should operate in the same manner on all platforms
  – Clean: minimal changes to code between platforms

• Reality:
  – Hacks and Tricks required to get adequate performance on multiple platforms:
    * Implement special 2D projections: 3D objects collapsed to 2D.
    * Implement (by hand) poor person’s Level of Detail: “small” objects represented with less detail
    * Solid objects (optionally) represented as line sets
    * Use selective visibility to keep number of objects to a minimum
  – Generic PHIGS too slow except on fast graphics systems:
    Sparc Ultra 3D Creator with G5G PHIGS \( \simeq \) HP 712 with HP PHIGS
  – vendor-specific PHIGS extensions are not uniform